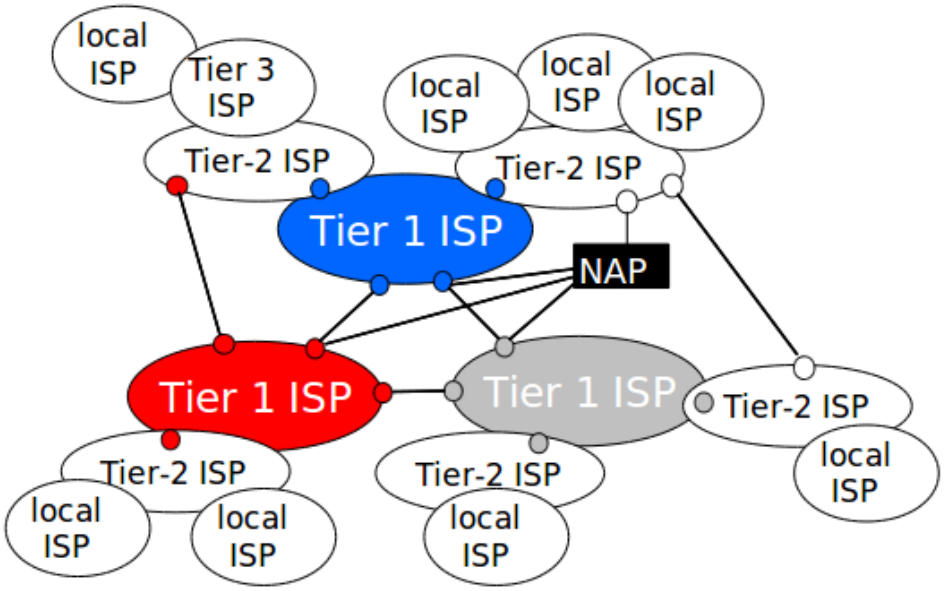
Hoofdstuk 1: Inleiding

1. **Bespreek de structuur van het Internet als "network of networks".**
   Internet is een gelaagde hiërarchie van aan elkaar gekoppelde netwerken. De backbone bestaat uit een internationale groep tier-1 ISP's (Verizon, AT&T) die allen onderling aan zeer hoge snelheid verbonden zijn. Ze verlenen elk toegang aan een groot aantal tier-2 ISP's (regionaal/nationaal) via peering die hun diensten dan weer verkopen aan andere ISP's en particuliere gebruikers (access ISP's).



2. **Leg uit wat een protocol is.**
   Een protocol is een afgesproken manier om te communiceren tussen twee of meerdere personen. Er worden afspraken gemaakt over de volgorde en formaat van boodschappen, antwoorden daarop en welke acties er aan berichten gekoppeld zijn. Voorbeeld: IRC Protocol, PING/PONG in IRC protocol.

3. **Geef de verschillende lagen van het TCP/IP referentiemodel. Geef bij elke laag een voorbeeld (leg kort uit).**
   Van boven naar onder:
   - Application Layer: applicatie-naar-applicatie communicatie. Dit zijn de toepassingen die verschillende netwerkfuncties implementeren. (FTP, HTTP)
   - Transport Layer: deze laag stuurt boodschappen van de applicatielaag op de host naar een applicatielaag op een andere host. (TCP, UDP)
   - Network Layer: stuurt pakketten van host naar host met behulp van het IP-protocol, en verschillende routeringsprotocollen.
   - Link Layer: verstuurt pakketten van de ene node naar de andere. Afhankelijk van het gebruikte protocol worden verschillende services aangeboden (bvb. reliable-delivery). Protocollen: Ethernet, WiFi, Point-to-Point protocol (PPP).
   - Physical Layer: staat in voor het uitwisselen van bits tussen twee nodes. Afhankelijk van het protocol gebruikt in de Link Layer. Als in de Link Layer bijvoorbeeld het Ethernet protocol gebruikt werd, zorgt deze laag voor implementaties voor Fiber, Coax etc.

4. **Bespreek de algemene werking van FTP.**
   Het FTP protocol gebruikt in de transport laag TCP.
   Er worden twee TCP connecties gebruikt
   - control connection (poort 21): voor het inloggen en uitvoeren van commandos

- data connection (poort 20): voor het versturen van de bestanden. Voor elk bestand wordt een nieuwe connectie geopend.

5. **Bespreek algemeen de eigenschappen van TCP en van UDP. Leg uit en vergelijk.**

TCP	*UDP*
Connection Oriented (state)	Connectionless (stateless)
3 Way handshake protocol	Eenvoudig protocol (sneller)
2 richtingsverkeer	1 richting
Stuurt segments	Stuurt datagram paketten
Flow & congestion control[1]	Geen van beide
Gegarandeerde sequentiële aflevering	Geen garanties
Vb. HTTP, FTP	Vb. Games, realtime toepassingen

6. **Bespreek de algemene werking van het internetprotocol (IP) en de typische eigenschappen.**
   Routeert berichten van de ene host naar de andere op basis van een unieke cijfercombinatie, het internetadres.
   Eigenschappen: best effort, enkele richting, conectionless

7. **Bespreek het principe van encapsulatie.**
   Encapsulatie laat toe dat de verschillende lagen onafhankelijk van elkaar werken. De applicatielaag stuurt een pakket (bvb FTP pakket) naar de Transport Layer. Elke laag neemt het pakket van de bovenliggende laag volledig over als payload en voegt zelf enkele headers toe vooraleer het zelf opnieuw door te geven.

8. **Leg het verschil uit tussen een host en een router.**
   - Host: eindsysteem (bvb webbrowser, mailserver, broodrooster), verwerkt pakketten tot en met de applicatielaag.
   - Router: dit zijn tussenpunten in een netwerk (packetswitches), verwerkt pakket maar tot in de netwerk-laag. Routers wachten tot ze een volledig pakket hebben ontvangen alvorens het wordt doorgestuurd (store-and-forward)

9. **Leg uit : client en server laag. Geef een voorbeeld. Vergelijk met client/server bij applicaties.**
   Elke laag van de netwerk-stack vervult een client- en/of server-functie aan de onderliggende/bovenliggende laag. Een clientlaag zal informatie doorgeven aan een onderliggende serverlaag die deze met bepaalde servicegaranties zal doorgeven.
   Bvb: TCP biedt gegarandeerde aflevering aan aan de bovenliggende app. laag.

   Vgl. met applicaties: server biedt diensten aan client-applicaties (bvb. mail)

10. **Bespreek identificatie in de applicatie-, transport- en netwerklaag.**
    - Applicatielaag: poort
    - Transportlaag: protocolnummer (6 = TCP, 17 = UDP)
    - Netwerklaag: IP-adres

11. **Hoe noemt men de informatieblokken in de applicatie-, transport-, netwerk- en datalinklaag ?**
    - Applicatielaag: message/bericht
    - Transportlaag: segment (datagram, enkel bij UDP)
    - Netwerklaag: (IP-)datagram
    - Linklaag: frame/fragment

12. **Wat is : IETF, RFC, ISP ? Geef kort uitleg.**
    - IETF, Internet Engineering Task Force: ontwikkelaars van internetstandaarden.
    - RFC, Request for Comments: documenten die de IETF standaarden beschrijven.
    - ISP, Internet Service Provider: Internet boer.

[1] Voorkomt belasting van de ontvanger en het netwerk.

Hoofdstuk 2: Applicatielaag

1. **Bespreek het client-server principe, op applicatieniveau.**
   De server biedt een service aan de client aan (bvb webserver -> webclient).
   Servers luisteren passief op een vast IP-adres en poort (bvb 80 voor http) en kunnen meerdere clients bedienen. Clients zetten actief een verbinding op met een server, hiervoor gebruiken ze een willekeurige poort.

2. **Bespreek het concept van "threads". Geef een voorbeeld.**
   Threads laten toe om verschillende zaken tergelijkertijd uit te voeren in hetzelfde programma. Zo kan een server luisteren op een bepaalde poort, om dan per inkomende connectie een aparte thread te openen om daarmee te communiceren.

3. **Welke transportdiensten kan een applicatie vereisen? Geef enkele voorbeelden.**[2]
   Er zijn 4 categorieën:
   - betrouwbaar (geen dataverlies)
   - throughput (bandbreedte nodig vs. elastisch)
   - tijdsgevoelig
   - beveiligd

   Voorbeelden:
   - webradio: kan tegen (beperkt) dataverlies, heeft veel bandbreedte nodig
   - games: tijdsgevoelig
   - filesharing: kan niet tegen dataverlies, gebruikt elastisch veel bandbreedte

4. **Bespreek het HTTP protocol en de belangrijkste protocolboodschappen.**
   HTTP: Hypertext Transfer Protocol
   Gebruikt TCP als onderliggend transport protocol. De client stuurt een HTTP request naar de server, die antwoordt met een HTTP response.
   HTTP Messages bestaan uit een request/status line, meerdere header lines, een lege lijn en dan de body. Er zijn twee soorten messages:
   - Request message:
     Request/Status line ziet er als volgt uit: "Method url version"
     Beschikbare methoden:
     - GET: opvragen van een URL
     - POST: invullen van een form, body bevat nieuwe informatie (kan ook via GET)
     - HEAD: zelfde als GET, maar enkel de headers worden teruggestuurd
     - PUT: uploaden van een object
     - DELETE: verwijderen van een object
   - Response message:
     Request/Status line ziet er als volgt uit: "[version] [statuscode] [phrase]"
     Status codes en phrases zijn bijvoorbeeld "200 OK" of "404 NOT FOUND"

5. **Waarvoor staat : HTTP, URL, HTML? Geef kort uitleg.**
   - HTTP, Hypertext Transfer Protocol: uitleg ↑
   - HTML, Hypertext Markup Language: opmaaktaal voor webpaginas
   - URL, Uniform Resource Locator: adres van een document

6. **Bespreek de verschillende HTTP connectiemogelijkheden.**
   - Non Persistent (default in HTTP/1.0)
     Elk opgevraagd object gebruikt een nieuwe TCP verbinding
   - Persistent (default in HTTP/1.1)
     Server laat de TCP connectie open na het verzenden van een object. Dit laat toe dat die later hergebruikt wordt. Na enige tijd inactief te zijn wordt de connectie gesloten. Pipelining laat toe meerdere requests na elkaar te versturen via dezelfde connectie zonder te wachten op een response.

[2] Internet biedt TCP en UDP aan.

7. **Bespreek een eenvoudig model voor responstijd bij HTTP.**
   Zonder persistentie: 2 keer de RTT[3] plus de tijd van het versturen van het object.
   Met persistentie: 1 keer de RTT voor het aanmaken van de connectie geteld worden.
   Met pipelining is de totale tijd ongeveer 2 keer de RTT plus de tijd voor het versturen van alle objecten.

8. **Bespreek het principe van cookies.**
   Cookies laten webservers toe om met state te werken in het stateless HTTP-protocol. Zo kunnen gebruikers een sessie hebben op een website (bvb op mailsite).
   HTTP paketten bevatten een header regel voor de cookie informatie die de client bij elke request meestuurt en die server ook in elk antwoord zal opnemen. Aan de hand van de informatie in de cookie kan een server een bezoeker “herkennen” en de bijbehorende informatie in een databank opzoeken.

9. **Bespreek conditional GET en waarvoor is dat nuttig?**
   Een conditional GET bestaat uit een HTTP request die de GET methode oproept, en een If-modified-since lijn heeft in de headers. Hierdoor worden objecten enkel teruggestuurd als ze niet meer gewijzigd zijn sinds de gespecifieerde datum.
   Het nut hiervan is dat bandbreedte uitgespaard wordt.

10. **Geef een overzicht van de verschillende e-mail protocols (afkorting + korte uitleg wat het doet).**
    SMTP: Simple Mail Transfer Protocol, versturen van mail tussen 2 hosts (push)
    POP: Post Office Protocol, ophalen van berichten bij een server (pull)
        3 fases: authorisatie, transactie, update
    IMAP: Internet Message Access Protocol, geavanceerdere versie van POP (heeft bijvoorbeeld mappenbeheer)
    RFC822/MIME: formaat van berichten en bijlages. RFC822 gaat over plain text en MIME over afbeeldingen, attachements etc.

11. **Bespreek het gebruik van naam en adres. Geef een voorbeeld.**
    Adres: cijfer van 4 bytes, vaste lengte en hiërarchisch interpreteerbaar
    Naam: leesbaar, makkelijk te onthouden, logische structuur

12. **Bespreek de DNS hierarchie. Geef een voorbeeld.**
    Lokale DNS-server: zoekt adressen op voor clients via andere nameservers
    Root name server: top level server, bevat adressen van alle TLD-nameservers
    Authoritative name server: originele host voor gegevens van een bepaald adres
    Intermediate server: tussenliggende server die gegevens van de authorative server gecached heeft liggen.

13. **Bespreek het iteratief en recursief mappen in DNS.**
    Iteratief mappen: wanneer een DNS-server niet verantwoordelijk is voor de gevraagde domeinnaam zal hij de nameserver van het gevraagde domein teruggeven aan de vraagsteller en zelf niet verderzoeken.
    Recursief mappen: de aangesproken DNS-server zal zelf recursief/iteratief verder het gevraagde domein opzoeken.
    In praktijk wordt een combinatie tussen de twee gebruikt. Eerst wordt iteratief bij de Root server het adres van de TLD nameserver opgevraagd. Daarna wordt vanaf de TLD nameserver recursief het adres van de eindbestemming opgevraagd.

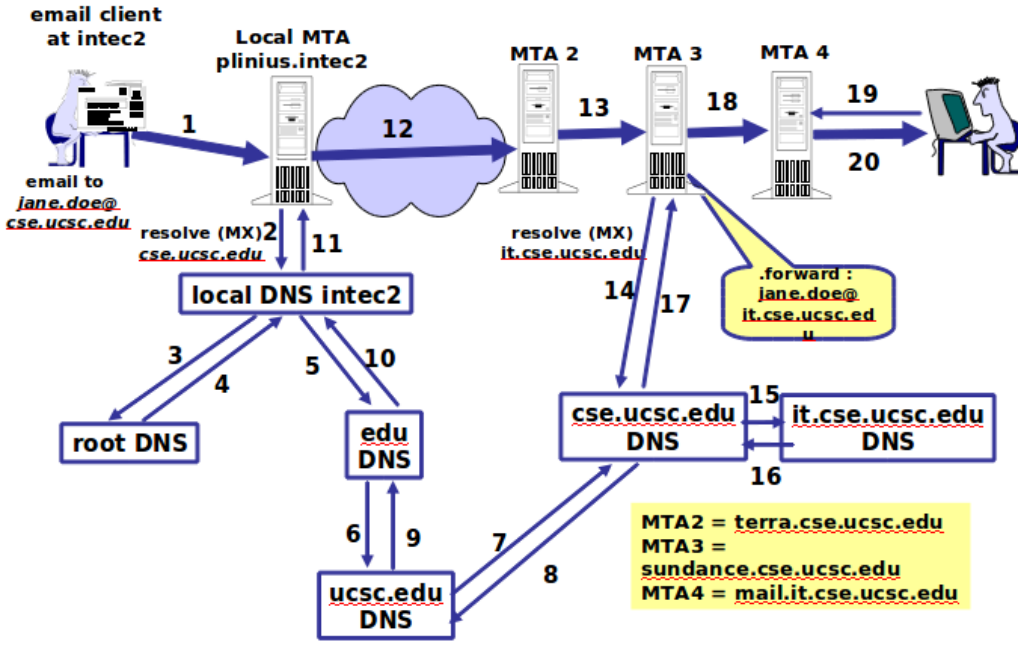
14. **Wat is (in de context van DNS) : RR, A, NS, CNAME, MX (geef ook een voorbeeld)**
    Een Resource Record (RR) is een blok data die de mapping van hostname op IP aanbied.
    In een DNS reply zit een Resource Record. Resource Records bestaan uit een 4-tuple (Name, Value, Type, TTL). De informatie wordt geïnterpreteerd afhankelijk van de waarde van het Type veld.
    - A-record: hostname op IP mapping, (hello.foo.com, 14.15.16.17, A, …)

[3]Round Trip Time, tijd van A naar B en terug.

- NS-record: mapt een domain op zijn authoritive DNS server, (foo.com, dns.foo.com, NS, …)
- CNAME-record: mapt een alias hostname op zijn canonical host name, (foo.com, server1.bar.foo.com, CNAME, …)
- MX-record: hostname van de mail-server die bij deze naam hoort, (foo.com, mail.bar.foo.com, MX, …)

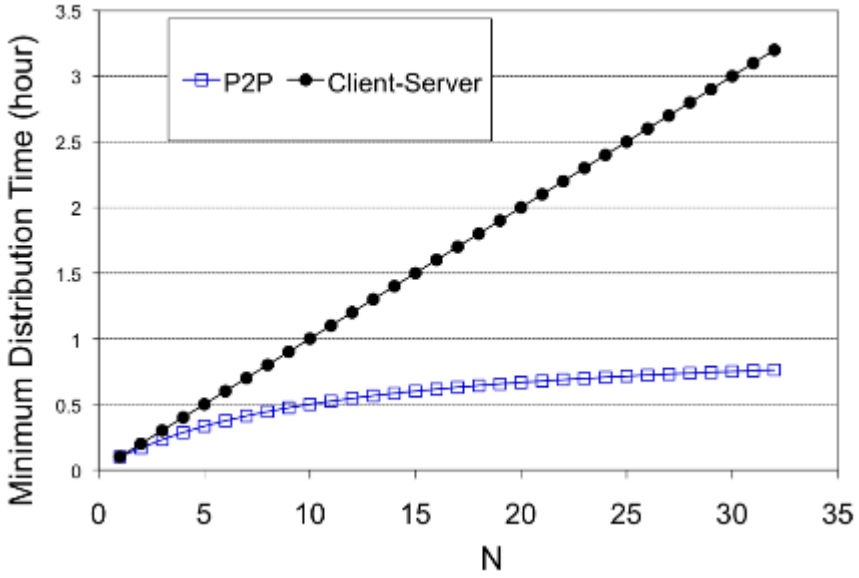
15. **Op het examen wordt een voorbeeld gegeven (b.v. MIME header, HTML file, DNS request) en er wordt gevraagd dat te bespreken.**

16. **Bespreek het voorbeeld DNS + e-mail (op het einde van de paragraaf over DNS). De figuur wordt opgegeven.**



1. De email wordt gepushed naar de lokale mailserver (MTA)
2. MTA doet een DNS request (MX) om cse.ucsc.edu te resolven.
3, 4. De lokale DNS server vraagt aan de root server het adres van de .edu TLD DNS
5-10. De lokale DNS server vraagt aan de .edu TLD DNS naar het adres.
11. De lokale DNS server stuurt het gevonden adres naar MTA.
12. MTA pusht de email naar MTA2
13. MTA2 pusht de email naar MTA3
14-17. MTA3 stuurt een DNS query om het adres van it.cse.uscs.edu te kennen
18. De mail wordt gepusht naar MTA4
19. De ontvanger vraagt zijn mail op
20. MTA4 stuurt de opgevraagde mail naar de ontvanger.

17. **Leg de figuur uit die de prestatie van C/S en P2P vergelijkt (de figuur wordt opgegeven).**



De grafiek geeft weer hoe lang het duurt om een bestand (grootte F) te verspreiden over N peers. We zien dat voor een Client-Server applicatie de functie lineair stijgt. Dit komt omdat de peers niet helpen bij het verspreiden van het bestand. Bij de P2P applicatie zien we dat ongeacht het aantal peers het downloaden van het bestand minder lang duurt dan bij de Client-Server applicatie. Als het aantal peers stijgt vlakt de functie af, dit is omdat peers ook uploaden naar andere peers

18. **Bespreek het principe van DHT.**
DHT: Distributed Hash Table
Dit zijn simpele databases die search en update operaties toelaten. Er worden key-value paren in opgeslagen. Peers kunnen dus via een key een value opvragen, of een nieuw paar in de database steken.
Voorstelling: circulair gelinkte lijst, met eventueel binnenwegen
Werking:
    - Elke peer krijgt een nummer uit $[0, 2^n - 1]$
    - Keys zijn ook nummers uit datzelfde domein. Via een hashfunctie kan een niet numerieke string omgezet worden. Als het ID-nummer van een key te groot is dan moet $id \ mod(2^n)$ beschouwd worden.
    - Het key-value paar wordt bijgehouden op de peer met het nummer dat gelijk aan, of dichtst bij (bijvoorbeeld het eerstvolgende) de waarde van de (gehashte) key is.
    - Als een peer wegvalt (peer churn) moeten de succesors geupdate worden.

Hoofdstuk 3: Transportlaag

1. **Bespreek multiplexering (connection-oriented and connectionless)**
Multiplexing laat toe om de Host-to-Host service van de netwerklaag uit te breiden naar een Proces-to-Proces service.
Multiplexing: bij de verzender, data van sockets wordt verzameld en in de header gestopt
Demultiplexing: bij de ontvanger, de ontvangen segmenten worden in de juiste socket geduwd.
Connectionless (UDP): de verbinding wordt gedefinieerd door het 2-tuple (destination IP, destination port). Alle connecties naar dezelfde poort komen dus ook bij hetzelfde socket/proces terecht.
Connection-oriented (TCP): de verbinding wordt gedefinieerd door het 4-tuple (source IP, source port, destination IP, destination port). Verbindingen met verschillende oorsprong,

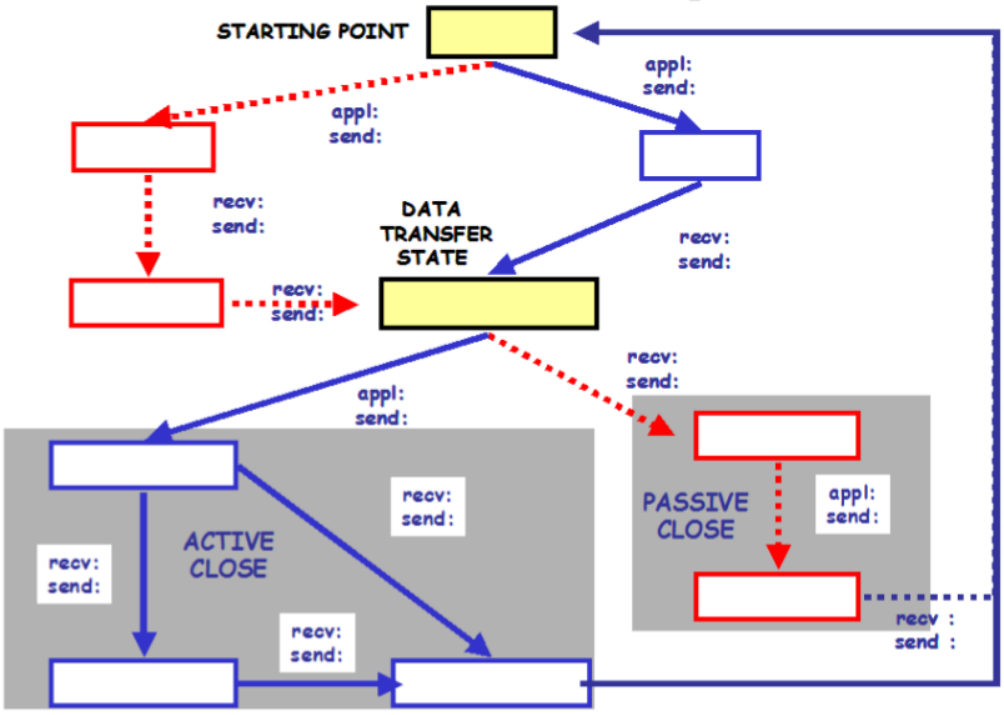
komen dus bij verschillende sockets terecht.

2. **Bespreek de verschillende velden van een TCP segment (het segment zelf wordt gegeven op het examen).**

16-bit source port number								16-bit destination port number
32-bit sequence number								
32-bit acknowledgement number								
4-bit header length	unused (6 bits)	URG	ACK	PSH	RST	SYN	FIN	16-bit window size
16-bit TCP checksum								16-bit urgent pointer
Options (if any)								
Data								

   - source & dest port numbers: worden gebruikt voor demultiplexing
   - sequence number: dient voor reliable data transfer. Bij de start wordt willekeurig een nummer gekozen om conflicten voorkomen, daarna het nummer van de eerste byte in het segment bij optellen.
   - ack number: het volgende sequence number dat men verwacht te ontvangen
   - header length: lengte van de header in 32bit woorden, dit kan varieren door eventuele opties.
   - flags: RST, SYN en FIN dienen voor opzetten van de connectie, ACK voor te acknowledgen van ontvangen data, URG voor dringende data aan te geven, PSH dat de data rechtstreeks aan de bovenliggende laag moet doorgegeven worden
   - urgent pointer: locatie van de laatste URG byte
   - checksum: voor error correcting
   - options: optionele info, kan dienen voor MSS te bepalen, of timestamping.
   - data: usually porn

3. **Bespreek het TCP toestandsdiagramma (een skeletfiguur wordt opgegeven, zonder enige tekst). Maak bij de uitleg eveneens gebruik van een tijdsverloop (tijdsas client- en serverzijde aangeven en welke boodschappen er uitgewisseld worden).**



4. **Bespreek : acknowledgment, timeout retransmit, duplicate reception, piggybacking, delayed ack, accumulated ack, selective retransmit, fast retransmit, retransmission timer, retransmission time-out, measured round trip time.**
   - Acknowledgement: bevestiging van correct ontvangen data

- Timeout retransmit: wanneer er na een bepaald interval geen ACK wordt ontvangen zal met opnieuw vesturen
- Duplicate reception: als de data correct toekomt bij de ontvanger, maar diens ACK gaat verloren zal de data opnieuw verstuurd worden, en dubbel ontvangen worden (en dus genegeerd)
- Piggybacking: ACK bit meezenden met een andere data die men wou versturen
- Delayed ack: enkele momenten wachten om een ACK te verzenden na het ontvangen van data
- Accumulated ack: ipv. meerdere ACKs te sturen, kan men 1 ACK sturen voor alle seqnumber < ack number
- Selective retransmit: wanneer in een hele reeks van segmenten, slechts 1 verloren gaat, zal enkel dit segment opnieuw verzonden worden zonder al de opvolgende.
- Fast retransmit: als met 3 duplicate ACKs (4 in totaal dus) ontvangt over 1 segment, zal men niet wachten op de time out om opnieuw te versturen
- Retransmission timer: timer voor het ontvangen ACKs, als deze afloopt versturen we opnieuw.
- Retransmission time-out: Het timeout interval berekend van EstimatedRTT en DevRTT, zie hieronder.
- Measured round trip time: laatst gemeten RTT, dit is de tijd tussen het verzenden van het segment en het onvangen van een ACK ervan.

5. **Hoe berekent men de RTO ? En hoe meet men de round trip time M ?**
   Voor elk verzonden segment (geen geretransmit) wordt de RTT gemeten, er kan wel maar 1 meting op hetzelfde moment gebeuren (SampleRTT). TCP houdt een gemiddelde van deze waarden bij adhv de formule:
   $$EstimatedRTT = (1-\alpha)*EstimatedRTT + \alpha*SampleRTT$$
   Waarbij de laatste waarden meer gewicht hebben dan oudere, voor α wordt best 0.125 gekozen. (EWMA = exponential weighted moving average)
   De gemiddelde afwijking ten opzichte van dit gemiddelde wordt gegegeven door:
   $$DevRTT = (1-\beta)*DevRTT + \beta*|SamplteRTT - EstimatedRTT|$$
   De RTO wordt dan berekend door
   $$RTO = EstimatedRTT + 4*DevRTT$$

6. **Bespreek het principe van flow control in TCP. Leg in detail uit a.d.h.v. diverse "windows". Waarom wordt flow controle gebruikt ?**
   Om te voorkomen dat de buffer aan de ontvangerszijde overloopt (en er dus paketten verloren gaan die later opnieuw verzonden zullen moeten worden), kan de zender zijn zendsnelheid aanpassen aan de snelheid waarmee de applicatie uit de buffer leest.
   Dit gebeurt door de zender een *send window* te laten bijhouden, die gebruikt wordt om bij te houden hoeveel paketten er nog verzonden kunnen worden zonder overflows. De ontvanger houdt op zijn beurt een *receive window* bij, met het aantal vrije plaatsen in zijn buffer.
   Send window = RcvWindow - (LastByteSent - LastByteAcked)
   Als de buffer volzit en alle paketjes al geACKed zijn, zal de zender niet op de hoogte gebracht worden wanneer er opnieuw plaats is. Daarom zal deze op regelmatige tijdstippen 1 Byte gaan versturen, deze zullen geAcked worden door de ontvanger. Zo zal een update van het rwnd doorgegeven worden.

7. **Bespreek het principe van congestion control in TCP. Waarom wordt dit gebruikt ?**
   Congestion control in TCP maakt gebruik van de volgende principes:
   - Slow Start
   cwnd = 1 en wordt met 1 verhoogt per ontvangen ACK (exponentieel stijgen)
   - Congestion Avoidance
   eenmaal voorbij een bepaalde limiet gaat het cwnd slechts stijgen met MSS/cwnd (lineair stijgen) = ongeveer 1 MSS per RTT

   Waarom: Om tot een stabiele toestand te komen waarin de connectie optimaal gebruikt wordt zonder het netwerk te overbelasten en weinig pakketjes te verliezen (dit brengt een kost met zich mee). Gezien de onderliggende netwerklaag hier niet

voor zorgt, neemt TCP deze taak op zich.

8. **Bespreek het verband tussen : send window, receiver window, congestion window**
   Send Window: minimum(RcvWindow, CongWindow) - sentButNotAckedBytes
   Receive Window: aantal bytes/MSS die we nog kunnen verzenden voor de ontvanger zijn buffer zal overflowen
   Congestion Window: aantal bytes/MSS die we nog kunnen verzenden voor de buffers van het netwerk/links/routers/mama's zullen overflowen
   Send window wordt dus beinvloedt door de andere 2, geen van beide mag overflowen

9. **Hoe wordt congestion gedetecteerd en wat is de reactie (geen details)?**
   In het geval van een timeout of triple duplicate ack gaat het congestion window halveren in grootte.
   Time-out: een pakket is verloren gegaan door congestion, heeft een te grote vertraging oplopen of is per ongeluk verloren gegaan (geen congestion, maar wordt zo geïnterpreteert)
   =>Slow start + Congestion avoidance.
   Quadruple ack: maar 1 pakket verloren gegaan, de andere zijn toegekomen (en genereren duplicate ACKs)
   =>Fast restransmit + fast recovery

10. **Bespreek het principe van slow start en congestion avoidance**
    Slow start: Wanneer een TCP connectie start is cwnd meestal 1 MSS, bij slow start zal voor elke ontvangen ACK deze waarde met 1 MSS opgeteld worden. Dit houdt in dat de cwnd elke RTT zal verdubbelen eigenlijk. Dit zal verdergaan tot er een pakket verloren gaat (time-out: sshtresh/2 en opnieuw beginnen, triple dupack: fast retransmit), of de waarde van *slow start treshold* bereikt wordt (vervolgens congestion avoidance)
    Congestion avoidance: Om te voorkomen dat we opnieuw congestion tegenkomen zal hier de waarde van cwnd met 1 MSS stijgen per RTT. Dit kan bereikt worden door cwnd te laten stijgen door MSS bytes elke keer een ACK toekomt.

11. **Bespreek het principe van fast retransmit en fast recovery.**
    Fast retransmit: als men 3 keer een duplicate ack ontvangt, zullen we er direct van uitgaan dat het segment verloren is gegaan en het opnieuw versturen, zonder te wachten op het aflopen van de timer.
    Fast recovery: de congestion window zal met 1 MSS verhoogd worden voor elke duplicate ACK die men ontvangen heeft voor het segment dat gezorgd heeft dat TCP dit stadium binnenging. (minimum 3 dus) Is niet verplicht door de RFC, maar wel aangeraden.

12. **Leg uit : AIMD.**
    Additive-increase Multiplicative decrease. De cwnd wordt lineair vergroot (+1) maar in het geval van een timeout wordt het gehalveerd. Zorgt voor 'saw tooth behavior' als men het in een grafiekje steekt

13. **Leg uit hoe TCP "fairness" ondersteunt. Geef een voorbeeld hoe men dat kan omzeilen.**
    Fairness: elke connectie krijgt een even groot deel van de beschikbare bandbreedte.
    TCP is fair omdat er telkens er een 3dup ack optreed het verschil in bitrate tussen twee TCP sessies wordt gehalveerd. Uiteindelijk zullen de connecties hierdoor convergeren naar een mooie verdeling.
    Dit kan omzeild worden door meerdere parallelle connecties te starten vanuit 1 proces, waardoor dit proces een groter deel zal krijgen.

14. **Waarvoor worden TCP en UDP gebruikt ? Geef enkele voorbeelden.**
    UDP: DNS, RIP, NFS, Internet telefonie, multimedia streaming
    TCP: SMTP, HTTP, FTP, dingen waar er betrouwbaarheid moe

Hoofdstuk 4: Netwerklaag

1. **Bespreek de verschillende IP-adresklassen. Geef enkele speciale adressen.**

A-klasse	eerste bit is 0	0-127.0.0.0/8	128 netwerken met 16mil adressen
B-klasse	eerste 2 bits 10	128-191.0.0.0/16	16k netwerken met 64k adressen
C-klasse	eerste 3 bits 110	192-223.0.0.0/24	2M netwerken met 256 adressen elk
D-klasse:	eerste 4 bits zijn 1110, gebruikt voor multicast		
E-klasse:	eerste 4 bits zijn 1111, gereserveerd		

   Speciale adressen:
   - Private networks: 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12
   - Loopback: 127.X.Y.Z (voornamenlijk 127.0.0.1)
   - Broadcast op eigen netwerk: 255.255.255.255
   - Broadcast op ander netwerk: X.Y.255.255
   - Source adres: 0.0.0.0

2. **Bespreek het principe van direct connected networks en subnetworks.**
   Direct connected: hosts die via dezelfde linklaag verbonden zijn (dus zonder router), kunnen elkaar direct aanspreken zonder forwarding.
   Subnetworks: concept op IP-laag dat mapt op direct connected networks. Alle hosts op een direct connected network hebben een identiek netwerk-gedeelte in hun IP-adres en hetzelfde subnetmasker.

3. **Bespreek subnet adressering. Geef een voorbeeld.**
   Een ip adres in een subnet bestaat uit 3 delen: netwerkgedeelte, subnetwerkgedeelte, en hostgedeelte. Bv 157.193.103.12 waarbij 157.193 het netwerk is, 103 het subnet, en 12 de host. De mask (gebruikt om de rand van het subnet en het host gedeelte aan te duiden) is hier dus FF.FF.FF.00.
   Subnet adressen met enkel 0'en of 1'en in het host gedeelte zijn niet toegestaan! Idem voor subnet gedeelte,

4. **Bespreek CIDR. Geef een voorbeeld.**
   CIDR: Classless Interdomain Routing
   Adressen worden opgedeeld in twee delen: een netwerk deel en een host deel. We stellen dit voor als a.b.c.d/x, waarbij x de lengte van het netwerk deel is in bits (bvb 12.13.14.0/23)
   Het netwerk deel heeft een arbitraire lengte.
   Voor CIDR was eer Classful addressing, dit resulteerde in de klassen uit vraag 1, en een ongelijke verdeling van de internet adressen.

5. **Bespreek het verschil tussen routing en forwarding.**
   Forwarding: wanneer een pakker arriveert in de router, moet die bepalen langs welke link interface hij het lokaal weer zal doorsturen
   Routing: de netwerklaag moet beslissen welke route/pad een pakket moet nemen over de gehele afstand tussen zender en ontvanger.

6. **Bespreek de verschillende velden van een IP-datagram (het datagram zelf wordt gegeven op het examen).**

   - Version: IPv4 of IPv6
   - Header length: Normaal 20 bytes, maar kan meer zijn door opties
   - Type of Service: bijvoorbeeld real time vs non real time.
   - Length: lengte van de header + data
   - Identification, flags & offset: om IP fragmentatie af te handelen
   - Time-to-live: tegengaan van eindeloos varen. Bij elke router wordt de TTL met 1 verminderd. Als de TTL 0 is wordt het pakket gedropt. De router stuurt dan een ICMP message (type 11 code 0) terug.
   - Protocol: Transport layer protocol, voor het verwerken bij de eindbestemming
   - Checksum
   - Source & Destination IP
   - Options: laat toe om extra info mee te zenden, wordt niet vaak gebruikt
   - Data: meestal transport layer segment

7. **Bespreek fragmentatie.**
   Omdat niet alle link-layer protocollen paketten (uit de network layer) van dezelfde lengte ondersteunen (adhv MTU = Maximum Transfer Unit) moeten ze gefragmenteerd worden. Hiervoor worden de identification, flags en fragmentation offset velden in een IP-datagram header gebruikt. Gefragmenteerde paketten hebben altijd dezelfde identification, maar met een verschillende ofset. Laatste pakket uit de reeks zal een fragmentation flag van 0 hebben.
   Om performantieredenen worden de gefragmenteerde paketten enkel op de end systems samengesteld, en niet op de routers tussenin.

8. **Wat is ICMP ? Geef een voorbeeld bij het gebruik in een redirect en traceroute.**
   ICMP: Internet Control Message Protocol, wordt door hosts gebruikt om controleberichten en foutmeldingen over het netwerk uit te wisselen. ICMP-berichten worden ingebed in een IP-datagram.

   Elk ICMP-bericht bestaat uit een type, een code, een checksum en een data sectie. Bij foutmeldingen zal de data sectie de IP-header en eerste 8 bytes van het pakket bevatten die de fout veroorzaakte. Voorbeeld: Type 3, code 1 = Destination host unreachable.

   Toepassingen:
   Redirect: Wanneer een host een bericht via R1 voor R2 verstuurt terwijl hij ook rechtstreeks verbonden is met R2 (zelfde subnet bvb.) zal R1 de host hiervan op de hoogte stellen met een een “Redirect”-bericht en zal de host zijn routing tables wijzigen.
   Traceroute: Wanneer de TTL van een IP-pakket 0 wordt zal de router waar dit gebeurt een “TTL expired”-bericht sturen naar de originele zender met zichzelf als afzender. Traceroute maakt hier gebruik van door IP-pakketten met incrementerende TTL’s te versturen en zo de route naar de bestemming te bepalen.

9. **Bespreek NAT. Geef een voorbeeld. Wat is large scale NAT ?**
   NAT: Network Address Translation.

Dit is een systeem waarbij meerdere hosts achter 1 publiek IP adres toch kunnen surfen. De router gedraagt zich tegenover de buitenwereld als een device met maar 1 ip adres, en verbergt de details van het privenetwerk. Er wordt gebruik gemaakt van een NAT Translation Table om een mapping te maken van poorten op de router, naar IP adressen en poorten binnen het privénetwerk.
Large scale NAT: wanneer ISPs aan NAT zouden doen (sup dawg, we heard you like NAT)

10. **Bespreek DHCP. Geef een voorbeeld.**
    DHCP: Dynamic Host Configuration Protocol.
    DHCP laat toe om dynamisch (plug & play) adressen te krijgen van een server. Hierdoor moet de network admin niet alle IP adressen zelf uitdelen. Dit is zeer handig in sitaties waar mensen met hun laptop verbinden in een netwerk. DHCP laat ook toe om IP adressen te hergebruiken (als een host offline gaat komt zijn IP adres vrij) of de lease erop te vernieuwen.
    Een client krijgt een adres in 4 stappen:
    - DHCP Server Discovery: broadcast op poort 67 om de server te vinden
    - DHCP Server Offer: broadcast van de server die antwoordt op de server discovery. Het verstuurde antwoord bevat o.a. een voorgesteld IP adres, en de tijd dat dit geldig is.
    - DHCP Request: de client kiest uit alle offers, en stuurt een request (met de configuratieparameters uit stap 2) naar de server van waar dit aanbod komt.
    - DHCP ACK: De server bevestigt dat verzoek.

11. **Wat is een AS ? Geef 3 types (waarom is het belangrijk een onderscheid te maken).**
    AS: Autonomous System
    Het internet bestaat uit verschillende Autonomous Systems die onderling vebronden zijn.
    - Stub AS: kleine onderneming, 1 verbinding naar andere ASs
    - Multihomed AS: grote onderneming, meerdere verbindingen naar andere ASs
    - Transit AS: provider, verbindt meerdere ASs

    Afhankelijk van het type AS wordt een ander routeringsalgoritme gebruikt, en alle routers in AS gebruiken hetzelfde routeringsalgoritme.

12. **Bespreek het verschil tussen intra- en inter-AS routering.**
    Intra-AS routing (ook interior gateway protocols) wordt gebruikt om te routes bepalen binnen een AS. Er worden voornamelijk twee routing protocols gebruikt: RIP (Routing Information Protocol) en OSPF (Open Shortest Path First).
    Inter-AS routing wordt gebruikt voor de routering tussen hosts in verschillende ASs. Hiervoor wordt BGP (Border Gateway Protocol) gebruikt.

13. **Bespreek het principe van distance vector en link-state routering. Geef een voorbeeld voor beide strategieën.**
    Distance Vector Routering: RIP is een Distance Vector protocol. Als metriek voor afstand wordt de hop count genomen (maximum 15, dus enkel kleine netwerken). Dit is het aantal subnetten op het pad.
    Elke 30 seconden wordt een een routing update naar de buren van de router gestuurd m.b.v. een RIP Response Message (ook RIP Advertisements). Hierin zitten maximum 25 subnetten in de AS, en de afstand van de afzender tot deze subnetten.

    Link-State routering: OSPF is een link state protocol. Elke router houdt de topologie van het netwerk bij in een zogenaamde Link State database. Hierop wordt Dijkstra's kortste pad algoritme uitgevoerd om het kortste pad naar elk subnet te bepalen, met het eigen subnet als root.
    Informatie over het netwerk wordt uitgewisseld via een broadcast naar het hele netwerk door middel van Link State paketten. Deze broadcasts gebeuren zowel periodiek (1 keer per 30 minuten) als op het moment dat een Link State wijzigt (bvb een wijziging in kost).

14. **Bespreek hierarchical OSPF. Waarom is dat nuttig ?**
    Hierarchische OSPF deelt een AS op in verschillende zones. Elke router stuurt dan link state broadcasts door naar routers in zijn zone. Om verbindingen tussen zones te

verwezenlijken zijn er Area Border Routers. Centraal in de AS staat één Backbone Area. De rol hiervan is het verkeer tussen verschillende zones in de AS te routeren. In de Backbone Area zitten alle Area Border Routers.
Het nut van hierarchical OSPF is dat, omdat de zones klein zijn, er minder rekenwerk gedaan moet worden.

15. **Bespreek een voorbeeld van BGP. Waarom heeft men I-BGP en E-BGP ?**
BGP: Border Gateway Protocol BGP wordt gebruikt voor Inter AS routing.
BGP biedt aan elke AS het volgende aan:
    a. Bereikbaarheids informatie over buur ASs
    b. Bereikbaarheidsinformatie doorsturen naar alle routers in het AS
    c. Goede routes naar subnetten bepalen.

    iBGP: Internal BGP, voor binnen één AS
    eBGP: External BGP, voor tussen twee verschillende ASs
    Waarom? met eBGP wordt routing informatie verdeeld tussen verschillende ASs, iBGP verdeelt deze informatie aan alle interne routers van het AS die dan zelf beslissen langs welke gateway router ze hun pakketten zullen routen.

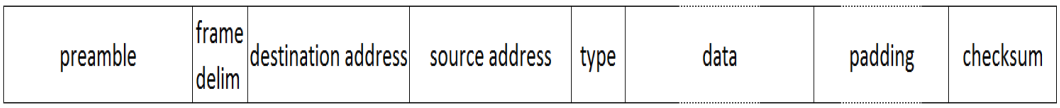
16. **Wat is een AS-PATH ? Wat is een NEXT-HOP ?**
AS-PATH: dit attribuut bevat het ASN (AS Number) van elke AS langswaar het advertisement is gepasseerd. Dus als een advertisement van A naar C gaat via B, voegt B zijn ASN toe. Dit voorkomt dat een router twee keer dezelfde advertisement stuurt.

    NEXT-HOP: dit is de router interface waar het AS-PATH begint.

17. **Wat is policy based routing in BGP ? Geef een voorbeeld.**
Policy based routing zijn politieke/economische regels die ervoor zorgen dat een router bepaalde routes niet zal adverteren, ook al bestaan deze paden. Een multihomed netwerk kan bijvoorbeeld geen transitverkeer willen en zal dan enkel de prefixen van zijn eigen netwerk adverteren.

Hoofdstuk 5: Datalinklaag

1. **Bespreek de verschillende velden van een Ethernet frame (het frame zelf wordt gegeven op het examen).**



Preamble: 7 bytes (10101010) die toelaten dat de adapter zijn klok synchroniseert met dat van de verzender.
Frame delimiter: 1 byte (10101011) die aangeeft dat wat volgt belangrijk is
Destination address: 6 bytes, MAC adres van de ontvanger.
Source address: 6 bytes, MAC adres van de verzender
Type: 2 bytes, soort data dat verzonden wordt
Data: eigenlijke data
Padding: extra nullen om aan de minimale frame length van 64 bytes te komen
Checksum: 4 bytes, CRC code voor fouten detectie

2. **Bespreek het CSMA/CD principe.**
CSMA: Carrier Sense Multiple Accese
CD: Collision Detection
Dit principe zegt dat twee regels gevolgd moeten worden
    1. Luister voor je zelf begint te transmitten (carrier sensing)
    2. Als iemand op hetzelfde moment begin te versturen, stop dan (collision detection)

3. **Waarom gebruikt men bij Ethernet een minimale framelengte van 64 bytes ?**
Anders zou het kunnen gebeuren dat er een collision optreed, maar omdat het segment

zo klein is zal het al geheel verzonden zijn eer het jam signaal de verzender bereikt. Hierdoor zal deze nooit doorhebben dat er een botsing was.(gaat wel tesamen met een beperking van de lengte van een ethernet lijn)

4. **Wat is exponential back-off (bespreek).**
Als er voor de n-de keer een jam signaal ontvangen wordt, zal de ontvanger voor K*512 bit tijden wachen, waarbij K willekeurig gekozen wordt uit {0,1,2,...,2^m-1} met m = min(n,10)
Dit wordt gebruikt omdat wanneer een node voor de eerste keer merkt dat er botsingen zijn, hij geen idee hoeveel andere adapters erbij betrokken zijn. Door K steeds te laten vergroten kan men zich aanpassen aan meerdere scenario's.

5. **Bespreek ARP bij Ethernet.**
ARP, Address Resolution Protocol: dit zorgt voor een vertaling tussen netwerk laag adressen (bvb IP adressen) en link laag adressen (MAC adressen).
Elke node heeft zijn eigen ARP table, die IP adressen op MAC adressen mapt. Elke mapping heeft een Time to Live die zegt wanneer ze vervalt (vaak 20 minuten). Als een node A (10.11.12.13) iets wil verzenden naar node B (10.11.12.14) dan heeft hij het MAC adres van B nodig. Als het MAC adres in de ARP tabel zit gebruikt hij dat. Als het er niet in zit gebruikt node A het ARP protocol om het MAC adres van B te bepalen:
    1. A construeert een ARP paket (met daarin o.a. zijn eigen IP en MAC adres)
    2. Dit paket wordt gebroadcast naar het hele netwerk
    3. B ontvangt dit paket en ziet dat zijn IP adres hierin zit. Hij beantwoordt met een nieuw ARP paket met de juiste mapping in.
    4. A ontvangt het pakket van B, en slaat de mapping op in zijn ARP tabel.

6. **Wat is een Ethernet hub ? En een Ethernet switch ?**
Hub: werkt op individuele bits ipv frames, elke bit die binnenkomt wordt versterkt en terug doorgestuurd naar alle andere interfaces van de hub.
Switch: Een switch zal een binnenkomend frame interpreteren, en het dan naar de juiste interface proberen doorsturen (als hij niet weet welke, naar allemaal). Er zal dus gefilterd worden, itt bij de hub. Switch interfaces hebben bovendien ook buffers, waarin de frames kunnen opgeslaan worden. Hij is echter wel compleet onzichtbaar voor de andere nodes, deze zullen niet op de hoogte zijn of hun paketten door switches zijn gepasseerd of niet.

7. **Hoe worden de switchtabellen ingevuld ? En hoe worden ze gebruikt ?**
Switchtabellen worden gebruikt voor filtering en forwarding. Entries in een switch tabel bevatten een MAC adres van een node, de interface naar die node en een timestamp.
Vullen: Switch tabellen worden automatisch gevuld, ze zijn zelf-lerend en worden als volgt gevuld:
    a. Initieel is de switch tabel leeg.
    b. Voor elk inkomend frame slaat de switch het MAC adres van de afzender op, de interface waarop het frame is binnengekomen en de huidige tijd op.
    c. Als na lange tijd geen frames van een bepaald MAC adres uit de tabel meer binnenkomen dan wordt de entry eruit gehaald.

    Gebruik: Stel dat een frame met destinationadres B binnenkomt via interface x. er zijn drie mogelijkheden:
    a. Er is geen entry voor B in de tabel. De switch forward het frame naar alle interfaces behalve X.
    b. Er is een entry die B op interface X mapt, de switch filtert (dropt) het frame.
    c. Er is een entry die B associeert met interface Y (Y != X), de switch forward het frame naar interface Y.

8. **Bespreek STP en geef een voorbeeld.**
STP, Spanning Tree Protocol: voorkomt dat een frame eindeloos het ruime sop bevaart door een opspannende boom van het netwerk op te bouwen.
De configuratie werkt als volgt:
    a. Blokeer alle poorten
    b. Kies een root switch, aan de hand van het Bridge ID (laagste eerst)

Bridge ID: 2 bytes Bridge Priority | 6 bytes Mac Adres
c. Maak een MST via Kruskal
d. Open poorten afhankelijk van de opspannende boom

STP is enkel bruikbaar in kleine LAN omgevingen omdat de recovery tijd hoog is (30 tot 60 seconden). RSTP of Rapid STP is een uitbreiding die een lagere recovery tijd heeft.

9. **Wat is een VLAN ? Bespreek twee types VLAN.**
VLAN, Virtual Local Area Network. Hosts in een VLAN kunnen met elkaar communiceren alsof ze (en geen enkele andere hosts) verbonden waren met de switch.

Port-based VLAN: de poorten van de switch worden in groepen verdeeld door de netwerk manager. In de praktijk krijgen poorten een VLAN ID toegewezen en kunnen ze enkel communiceren met poorten met hetzelfde VLAN ID. Verkeer tussen verschillende VLAN's gebeurt via een apparte router.

Tagged VLAN: frames krijgen een extra tag header. In de header zit informatie over het VLAN van waar het komt. Switches en end stations zijn VLAN-aware als ze tagged frames ondersteunen.
Frames zonder tag zijn untagged frames. Er zijn twee soorten tagged frames:
a. VLAN tagged frame: tagged frame met VLAN identificatie en priority informatie in de header.
b. Priority-tagged frame: tagged frame met enkel priority informatie (VLAN ID = 0)

10. **Geef een aantal voor- en nadelen van switches (versus routers).**
Switches:
+ Plug & play
+ Hoge filtering en forwarding rates (omdat switches maar 2 lagen implementern)
- Topologie van een netwerk beperkt zich tot MST
- Grote ARP tabellen in alle nodes
- Geen bescherming tegen broadcast storms

Routers:
- Heeft configuratie nodig
- Duurt langer om paketten te verwerken (3 layers)
+ Bied meerdere paden tussen twee hosts aan
+ Firewall bescherming tegen layer-2 broadcast storms

11. **Bespreek PPP.**
PPP, Point-to-Point protocol.
Dit protocol laat toe om een directe verbinding (vs. broadcast) tussen twee nodes te leggen.
Bij het ontwerp legde IETF enkele requirements op: simplicity, packet framing, transparantie (alle bit patterns toegelaten), ondersteuning van meerdere netwerk laag protocollen op dezelfde fysische link, ondersteuning voor meerdere link typen, error detectie, detectie van linkfalen.
PPP kan mogelijks nog nog andere services aanbieden: error correction, flow control, sequencing en multipoint links.

Hoofdstuk 8: Beveiliging

1. **Bespreek een aantal mogelijke aanvallen op het internet (en de bijhorende verdedigingen)**
MITM <- MAN IN THE MIDDLE DEB
Mapping: via port scans (proberen opzetten van een TCP connectie op verschillende poorten) te weten welke poorten open staan en welke services er actief zijn.
Sniffing: Pakketten die gebroadcast worden of over een broadcast-medium gestuurd worden kunnen gelezen worden door iedereen in het netwerk.
IP-Spoofing: het bronadres in een IP-pakket kan makkelijk ingevuld worden met een valse waarde. Oplossingen: routers filteren pakketten met onmogelijk origine,

authenticatie van IP-pakketten adhv IPsec.
(Distributed) Denial of Service: door een grote hoeveelheid pakketten van 1 of meerdere machines naar 1 host te sturen kan zijn werking stilgelegd worden aangezien hij voor elk pakket bepaalde resources moet alloceren. Oplossingen: evil pakketjes wegfilteren, bron zoeken van pakket en daar het probleem 'oplossen'.

2. **Bespreek het principe van cryptografie met symmetrische sleutels.**
Beide partijen gebruiken dezelfde sleutel om een boodschap te versleutelen en ontsleutelen. Deze sleutel is enkel door hun gekend.
Voorbeeld:
Persoon A schrijft de boodschap "Gilles is demax" (plaintext)
De boodschap wordt versleuteld met sleutel K en is nu "G!ll3s !s d3m4x" (ciphertext)
Deze versleutelde boodschap wordt verstuurd over het internet.
Persoon B kan deze boodschap nu ontsleutelen met sleutel K, en lezen.

3. **Bespreek het verschil tussen blok- en stroomversleuteling. Wat is CBC ?**
Stroomversleuteling: er wordt 1 bit per keer versleuteld

Blok versleuteling: de volledige boodschap wordt opgedeeld in blocks, elk blok wordt appart versleuteld. Er wordt een 1 op 1 mapping gemaakt om een blok van k bit te versleutelen. Als k echter groot (bvb. 64) is worden de tabellen met de mappings gigantisch. Als oplossing hiervoor wordt een prototype functie gebruikt. Die zal het blok opgdelen in stukken van 8 bit, die appart versleuteld worden. Hierna worden de blokken van 8 bit achter elkaar geplakt, en wordt het resultaat gepermuteerd. De prototype functie wordt meer dan eens uitgevoerd.

CBC, Cipher-Block Chaining:
Als een twee identieke blokken versleuteld worden produceren ze dezelfde output. Om dit tegen te gaan worden RNG's gebruikt. De verzender vraagt een willekeurig k-bit getal (R) op voor elk blok (B). Het versleutelde blok (E) is dan $E = K_s(B\ xor\ R)$. De verzender stuurt nu $E_1, R_1, E_2, R_2, \ldots$ Hoewel het willekeurige getal ook verstuurd wordt kan de boodschap niet vertaald worden omdat de hakker de sleutel niet kent.
Omdat nu twee keer zoveel bits verstuurd moeten worden wordt Cipher Block Chaining gebruikt. Enkel voor het eerste blok wordt nu een willekeurig k-bit getal gegenereerd. Voor de hierop volgende blokken wordt het het versleutelde vorige blok als k-bit getal gebruikt.

4. **Bespreek het principe van cryptografie met openbare sleutels.**
Elke persoon heeft nu twee sleutels. Er is een publieke sleutel die iedereen kan opvragen en een private sleutel.
Als A een boodschap naar B wil sturen vraagt A de publieke sleutel van B op en versleutelt de text daarmee. B kan dan zijn private sleutel gebruiken om de boodschap te ontcijferen.
Mogelijke problemen zijn dat de hakker boodschappen versleutelt met B's publieke sleutel en kijkt of delen van het bericht van A hierin voorkomen. Ook kan iedereen een bericht naar B versturen in naam van A.

5. **Bespreek het principe van digitale handtekening.**
Een digitale handtekening is een cryptografische techniek om aan te tonen dat iemand te auteur/eigenaar/creator van een document is.
B kan bij het versturen van een bericht M naar A het bericht hashen (h(M)) en die hash versleutelen met zijn private sleutel. A kan dan zelf een hash van het bericht berekenen, en die vergelijken met de hash die B versleuteld heeft (door het met B zijn publieke sleutel te ontcijferen).

6. **Bespreek KDC en CA.**
KDC, Key Distribution Center: laat toe om bij het gebruik van symmetrische sleutels tussen 2 partijen een gedeelde secret key te bepalen. Het KDC is een centrale server waarmee gebruikers communiceren via een gedeelde geheime sleutel. Persoon A kan aan het KDC de geheime sleutel van B opvragen. Het KDC gaat die sleutel terugsturen,

maar versleuteld met de unieke sleutel van A. A kan die gecodeerde sleutel nu decoderen en communiceren met B.

CA, Certification Authority: laat toe om te bevestigen dat een publieke sleutel wel degelijk van de juiste persoon is. Een persoon kan zijn publieke sleutel registreren bij de CA. Hiervoor moet die persoon wel zijn identiteit bewijzen aan de CA. Als dit lukt maakt de CA een certificaat dat de persoon linkt met zijn publieke sleutel. Als persoon B nu een bericht naar A stuurt voegt hij zijn certificaat toe. Persoon A kan nu de publieke sleutel van de CA gebruiken om de identiteit van B te controleren.

7. **Bespreek principe e-mail encryptie.**
Bij het versturen van emails zijn enkele security services gewenst: confidentiality, sender/receiver authentication en message integrity.
Om confidentiality te bereiken kan het volgende gedaan worden:
    1. Zender kiest een willekeurige sleutel
    2. Zender versleutelt het bericht met de willekeurige sleutel
    3. Zender versleutelt de willekeurige sleutel met de publieke sleutel van de ontvanger
    4. Zender verstuurt het verseutelde bericht en de versleutelde willekeurige sleutel
    5. Ontvanger ontcijfert de willekeurige sleutel en daarna de boodschap.

Message integrity en sender authentication kan bereikt worden door het bericht eerst te hashen, die hash te versleutelen met de private sleutel van de verzender en daarna te concateneren met het originele bericht.
Een populair algoritme is PGP (Pretty Good Privacy)

8. **Bespreek principe SSL ("toy example").**
SSL: Secure Socket Layer
SSL voegt data integrity, server/client authentication en confidentiality toe aan TCP.
SSL bestaat uit 3 stappen: handshake, key derivation en data transfer.

Stel dat B informatie naar A wil sturen:
Handshake:
    a. opstellen TCP connectie (TCP handhskake) tussen A en B
    b. Verificatie van A (via certificate)
    c. Doorsturen van een master key van B naar A

Key derivation: A en B gaan allebei 4 sleutels genereren:
    a. E_B encryptie sleutel voor data van B naar A
    b. M_B MAC sleutel voor data van B naar A
    c. E_A encryptie sleutel voor data van A naar B
    d. M_A MAC sleutel voor data van A naar B

Data transfer: de data wordt opgedeeld in records, waaraan een MAC (Message Authentication Code) toegevoegd wordt voor integrity checks

9. **Bespreek principe IPSec: twee modes, SA**
Via IPSec zal de gehele payload van het IP paket geëncrypteerd en/of geauthenticeerd worden, waardoor alle data uit de bovenliggende protocollen onzichtbaar zal worden (='blanket coverage'). Wordt gebruikt voor VPN.

2 modes: Transport en Tunnel mode. Bij transport wordt het IPsec datagram verstuurd en ontvangen door de 2 hosts. Bij tunnel zijn 2 routers (dit kunnen ook de hosts zelf zijn) op de hoogte van de IPsec verbinding en gebeurt dit volledig transparant voor de hosts, de IPsec transformatie wordt uitgevoerd op het volledige originele IP-pakket en in een nieuw IP-pakket verstuurd.

SA: vooraleer informatie via IPsec verstuurd kan worden dient er een virtuele simplex connectie opgezet te worden door de 2 eindpunten waarin alle beveiligingsinstellingen doorgegeven worden. IPsec is dus connectie-georiënteerd.

10. **Bespreek pakketfiltering "packet firewall: stateless en stateful" en toepassingsgateway "application gateway"**

Als een firewall rekening houdt met de openstaande TCP connecties om te beslissen of hij een pakket zal tegenhouden of niet, dan noemen we hem statefull. Dan wordt er ook gebruik gemaakt van een connection table, naast de lijst met algemene rules. Stateless firewalls werken pakket-per-pakket.

Application gateway: Hiermee kunnen we trafiek filteren adhv applicatie regels. De application gateway wordt als een tussenstop gebruikt om alle telnet/skype/whatever trafiek langs te routen, en enkel de pakketten die van deze source komen worden doorgelaten door de firewall. De applicaties moeten hiervoor wel weten hoe ze de app. gateway moeten contacteren.

Hoofdstuk 10: IPv6

1. **Bespreek de verschillende types adressen.**

Unicast (one-to-one)
- Link-local address (1 per interface): prefix FE80::/10
- Global address: prefix 2000::/3 + global routing prefix = /48
- Unique local address (for local communications): prefix FC00::/7

Multicast (one-to-many): prefix FF00::/8 + 4 flag bits + 4 scope bits
bvb: FF02::1 = alle nodes op het lokale netwerk

Anycast (one-to-nearest = one-to-one-of-many):
bvb. dichtsbijzijnde DNS-server op netwerk

2. **Wat zijn de belangrijkste verschillen tussen een IPv4 en IPv6 header ? Waarom heeft men die verschillen ingevoerd ?**

De grootte is verdubbeld naar 40 bytes (aangezien zender/ontvanger elk al 16 bytes zijn) maar is nu vast van grootte (makkelijker te verwerken). Het versie veld bevat nu 6 ipv 4, ToS is hernoemd naar Traffic Class, ipv. Total Length word enkel nog Payload Length aangeduid (aangezien de header toch vast is van grootte). Protocol-veld is hernoemd naar Next Header en TTL naar Hop Limit.

Verdwenen velden: header length (staat toch vast), checksum (was duur om telkens te herberekenen en bij moderne fiber connecties komt corruptie nog maar zelden voor), ID, flags en fragmentatie-offset (routers fragementeren geen pakketten meer!).
Toegevoegde velden: flow label om paketten te groeperen.

Er kunnen dus geen opties meer toegevoegd worden aan het IP pakket, in plaats daarvan dient men extensie headers toe te voegen waarnaar dan verwezen wordt vanuit Next Header. Voorbeelden van zo'n extensie headers zijn Routing headers, Fragment Headers. Afhankelijk van het type header worden deze enkel geïnterpreteerd door de bestemmeling of door sommige/alle tussenliggende routers (Hop-by-Hop en Routing headers).

3. **Geef een voorbeeld van adresresolutie.**

Tegenhanger van ARP in IPv4 is Neighbour Discovery Protocol (NDP) dat gebruik maakt van ICMPv6. Maakt gebruik van lokale multicast adressen en link-local adressen en sollicited-node multicast adressen (multicast adres afgeleid van laagste 24-bits van het gezochte adres, prefix FF02::1:FF00:/104).

Wanneer host A het link-layer adres van host B wil te weten komen stuurt hij een Neighbour Sollicitation naar het sollicited-node-multicast-adres gebaseerd op het adres van B (hierbij kan ook gebruik gemaakt worden van het Ethernet multicast-adres dat begint met 33-33-...). Er kan geantwoord worden met een Neighbour advertisement in unicast.

4. **Bespreek de verschillende autoconfiguratiestappen.**

Stap 1: Genereer een link-local address op basis van EUI-64 (MAC-adres) en word lid van de all-nodes en sollicited-node multicast groepen. Voer duplicate-address-detection (DAD) uit en leg vervolgens het link-local address vast als voorkeursadres.

Stap 2 (stateless): Router sollicitation (via router multicast), elke router antwoordt met Router Advertisement met netwerk prefixes. Voor elke prefix die via autoconfiguratie ingesteld dient te worden genereert de host een nieuw adres en controleert of dit nog vrij is via DAD.

Stap 3 (stateful): Wanneer netwerk prefixes uit de Router Advertisement de Managed/Other flag hebben dient een adres via DHCP verkregen te worden of kunnen er extra configuratiedetails daar opgevraagd worden.

5. **Geef de basisprincipes die kunnen gebruikt worden bij een overgang van IPv4 naar IPv6.**

Dual-stack = afhankelijk van de andere partij 'praat' de host IPv4 of IPv6. Dit vereist echter wel dat ook bvb. de ISP IPv6 ondersteunt en route. Een ander nadeel is dat alle configuraties dubbel moeten onderhouden worden (wat kan leiden tot conflicten/beveiligsproblemen), niet alle software is hiermee compatibel.

Tunneling = gebruik maken van de bestaande IPv4 infrastructuur om IPv6 pakketjes ter versturen. Vereist een tunnel die verbonden is met zowel IPv4 als IPv6 internet.

Translation = vertaal IPv4 adressen naar IPv6 adressen (en omgekeerd) door embedding/NAT-technieken, dit dient zo transparent mogelijk te gebeuren maar niet alles kan 1 op 1 met elkaar gemapped worden.